📘 Pet Insurance Analytics Portfolio Project

A full end-to-end SQL + Python + BI project simulating a real-world pet insurance business

🐾 Project Summary

This project simulates the operations of a UK-based pet insurance company, modelling customers, pets, policies, claims, and vet visits in a fully relational MySQL database. The data is synthetically generated using Python, with realistic business logic and referential integrity. The goal is to demonstrate the full analytics workflow — from data engineering and SQL analysis to business intelligence and insight storytelling — in a way that mirrors the expectations of a junior data analyst role.

🔍 Key Business Questions

This dataset enables analysis of real-world insurance KPIs and customer behaviour:

- Which pet breeds generate the highest average claim costs?

- How does pet age affect policy risk and claim frequency?

- What is the loss ratio by product and coverage type?

- What is the average customer lifetime value (CLV)?

- Which vet clinics are most expensive or most visited?

- How many customers own multiple pets or policies?

Insight findings are available here

🛠️ Tech Stack

- **MySQL** – relational database design and querying

- **Python** – data generation and scripting

- **Pandas / NumPy / Seaborn** – analysis and visualization

- **Tableau** – dashboards and reporting

- **GitHub** – version control and documentation

🗃️ Database Schema

The schema models the core entities of a pet insurance provider:

Table	Description
customers	Customer demographic and contact details
pets	Pet profiles linked to customers
products	Insurance products with pricing and coverage
policies	Policy subscriptions per pet
claims	Claims submitted under policies
vet_visits	Vet appointments linked to claims
invoice_line_items	Itemized treatment costs per visit
claim_payments	Payments issued for approved claims

⚙️ How to Run Locally

1. **Create the database**

```bash
mysql -u root -p < sql/create_db.sql
```

2. **Load the data**

```bash
mysql -u root -p < sql/batch_load_csv.sql
```

3. **Explore the data**

Run `sql/data_overview_queries.sql` to get a feel for the dataset.

4. **Run analysis**

Use the Jupyter notebooks in `/notebooks` or execute SQL from `analysis_queries.sql`.

5. **Export processed views to CSV**

```bash
cd scripts/
python export_views_to_csv.py
```

This will save full CSVs to `data/processed/` for use in dashboards or Python notebooks.

6. **Build dashboards**

Connect Power BI or Tableau to your processed CSVs or MySQL views and build visuals.